

05-25-00

JC828
05/23/00
U.S. PTO

Express Mail Label No. EL172582250

Docket No.: END9 1999 0129 US1

10789458215
A

**NEW UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)**

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Total Pages this Submission: 3

TO THE ASSISTANT COMMISSIONER FOR PATENTS

Box Patent Application
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:
SYSTEM AND METHOD FOR NETWORK ADDRESS TRANSLATION INTEGRATION WITH IP SECURITY

and invented by:

EDWARD B. BODEN, MARK J. MELVILLE, TOD A. MONROE, FRANK V. PAXHIA

If a **CONTINUATION APPLICATION**, check the appropriate box and supply the requisite information:

Continuation Divisional Continuation-in-part(CIP) of prior application
No.: 09/240,720 FILED 29 JAN 1999.

Enclosed are:

Application Elements

01. Filing fee as calculated and transmitted as described below
02. Specification having 50 pages and including the following:
 - a. Descriptive Title of the Invention
 - b. Cross References to Related Applications (if applicable)
 - c. Statement Regarding Federally-sponsored Research/Development (if applicable)
 - d. Reference to Microfiche Appendix (if applicable)
 - e. Background of the Invention
 - f. Brief Summary of the Invention
 - g. Brief Description of the Drawings (if drawings filed)
 - h. Detailed Description
 - i. Claim(s) as Classified Below
 - j. Abstract of the Disclosure
03. Drawing(s) when necessary as prescribed by 35 USC 113)
 - a. Formal
 - b. Informal

Number of sheets: 8

NEW UTILITY PATENT APPLICATION TRANSMITTAL

(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Total Pages this Submission:3

Application Elements (Continued)

04. Oath or Declaration

a. Newly executed (original or copy) Not executed

b. Copy from a prior application (37 CFR 1.63(d)) (*for continuation/divisional application only*)

c. With Power of Attorney Without Power of Attorney

05. Incorporation By Reference (*usable if box 4b is checked*)

The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Item 4b, is considered as being part of the disclosure of accompanying application and is hereby incorporated by reference therein.

06. Computer Program in Microfiche (*Appendix*)

07. Nucleotide and/or Amino Acid Sequence Submission (*if applicable, all must be included*)

a. Paper Copy

b. Computer Readable Copy (*identical to computer copy*)

c. Statement Verifying Identical Paper and Computer Readable Copy

Accompanying Application Parts

08. Assignment Papers (*cover sheet & document(s)*)

09. 37 CFR 3.73(B) Statement (*when there is an assignee*)

10. English Translation Document (*if applicable*)

11. Information Disclosure Statement/PTO-1449 Copies of IDS citation

12. Preliminary Amendment

13. Acknowledgment postcard

14. Certificate of Mailing

First Class Express Mail (Label No: EL172582250)

15. Certified Copy of Priority Document(s) (*if foreign priority is claimed*)

NEW UTILITY PATENT APPLICATION TRANSMITTAL**(Large Entity)**

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Total Pages this Submission: 3

Accompanying Application Parts (Continued)

16. Additional Enclosures (identify below)

Fee Calculation and Transmittal**CLAIMS AS FILED**

For	# Filed	# Allowed	# Extra	Rate	Fee
Total Claims	22	-20	2	x \$18.00	\$ 54.00
Indep. Claims	13	- 3	10	x \$78.00	\$ 780.00
Multiple Dependent Claims	_____				\$
				BASIC FEE	\$ 690.00
Other FEE (Specify purpose)					\$
				TOTAL FILING FEE	\$1,524.00

- A check in the amount of \$ to cover the filing fee is enclosed.
- The Commissioner is hereby authorized to charge and credit IBM Corporation Deposit Account No. 09-0457 as described below. A duplicate copy of this sheet is enclosed.
- Charge the amount of \$1,524.00 as filing fee.
 - Credit any overpayment.
 - Charge any additional filing fees as required under 37 C.F.R. 1.16 and 1.17.
 - Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

Dated: 23 MAY 2000


Signature
Shelley M Beckstrand
Reg. No. 24,886314 Main Street
Owego, NY 13827-1616
Phone: (607) 687-9913

cc:

transmitnew.wpd

CERTIFICATE OF MAILING BY "EXPRESS MAIL" (37 CFR 1.10)Applicant(s): **E. B. Boden et al**

Docket No.

END9-1999-0129US1

Serial No.

Filing Date
Herewith

Examiner

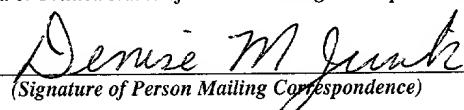
Group Art Unit

Invention: **System And Method For Network Address Translation Integration With IP Security**

1059/57825 pro

I hereby certify that this **New Patent Application, IDS w/References, Assignment w/Cover Sheet**
(Identify type of correspondence)

is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under
37 CFR 1.10 in an envelope addressed to: The Assistant Commissioner for Patents, Washington, D.C. 20231 on
May 23, 2000

*(Date)***Denise M. Jurik***(Typed or Printed Name of Person Mailing Correspondence)*

*(Signature of Person Mailing Correspondence)***EL172582250***("Express Mail" Mailing Label Number)*

Note: Each paper must have its own certificate of mailing.

APPLICATION
FOR
UNITED STATES LETTERS PATENT

APPLICANT NAME E. B. BODEN, ET AL
TITLE SYSTEM AND METHOD FOR
 NETWORK ADDRESS TRANSLATION
 INTEGRATION WITH IP SECURITY
DOCKET NO. END9 1999 0129 US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF MAILING UNDER 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C., 20231 as "Express Mail Post Office to Addressee" on May 23, 2000
Mailing Label No. EL172582250

Name of person mailing paper: Denise M. Jurik

Denise M. Jurik 5/23/2000
Signature Date

**SYSTEM AND METHOD FOR NETWORK ADDRESS TRANSLATION
INTEGRATION WITH IP SECURITY**

Background of the Invention

Cross References to Related Applications

5 This application is a Continuation-In-Part (CIP) of Serial No. 09/240,720, filed 29 Jan 1999, entitled "System and Method for Network Address Translation Integration With IP Security".

10 U.S. patent applications Serial No. 09/239,693, filed 1/29/99, entitled System and Method for Managing Security Objects; Serial No. 09/240,718, filed 1/29/99, entitled "System and Method for Dynamic Macro Placement of IP Connection Filters"; S/N 09/239,694, filed 1/29/99, entitled "System and Method for Dynamic Micro Placement of IP Connection Filters"; S/N 09/240,483, filed 1/29/99, entitled "System and Method for Central Management of Connections in a Virtual Private Network, are assigned to the same assignee hereof and contain subject matter related, in certain respects, to the subject matter of the present application.

15 The above-identified patent applications are incorporated herein by reference.

Technical Field of the Invention

This invention pertains to security over virtual private network (VPN) connections. More particularly, it relates to VPN NAT, or concurrent use of network address translation (NAT) and Internet Protocol (IP) Security (IPSec) protocols.

Background Art

Virtual Private Networks (VPNs) are an active area of technical development throughout the Internet and computing industry. This is because they are a fundamental enabler for most forms of e-business. VPNs use protocol tunneling and encryption and decryption technology (IP Security protocols) to allow clients and servers, branch offices, or independent organizations to exploit the Internet for their TCP/IP traffic at much lower costs than buying dedicated lines, without losing a key benefit of dedicated lines: privacy.

The tunneling that VPN employs has a side effect, which creates a problem: two subnets or companies, or other users, which didn't initially communicate directly, now do, and

this greatly increases the likelihood of IP address conflicts.

Network Address Translation (NAT) is widely deployed in Internet and in companies connecting to the Internet to overcome address conflicts. These conflicts commonly occur between designated 'private' address spaces (e.g. 10.*.*.*).

However, NAT and IP Security (IP Sec) are architecturally conflicting. In fact, NAT breaks IP Sec. That is, NAT "is the feature which finally breaks the semantic overload of the IP address as both a locator and the end-point identifier" (see, "Architectural Implications of NAT", draft-iab-nat-implications-00.txt, March 1998. IPSec is described in Kent, S., and Atkinson, "Security Architecture for the Internet Protocol", RFC2401, November 1998; Kent, S., and Atkinson, "IP Authentication Protocol", RFC 2402, November 1998; and Kent, S., and Atkinson, "IP Encapsulation Security Payload", RFC 2406, November 1998.) As a result, two hosts cannot establish an IP Sec connection if there is a NAT system in between. There are two reasons why. First, the IP traffic that flows between the two hosts

(for the IP Sec connection) will have authentication protocol (AH) or encapsulation security payload (ESP) applied. (See RFC's 2402 and 2406, *supra*.)

First, with respect to ESP in tunnel mode, the IP
5 address that needs to be translated is inside the ESP tunnel
and is encrypted. It is, therefore, unavailable to NAT.
With respect to AH in transport or tunnel mode, the IP
address that needs to be translated is visible in NAT, but
the AH authentication includes it. Therefore, changing the
10 IP address will break the authentication at the remote end
of the IP Sec connection. With respect to ESP in transport
mode, even if ESP is used with authentication, the IP
address is available to NAT. But, if the IP address is
changed, the IP Sec connection breaks due to the breaking of
15 authentication at the remote end of the IP Sec connection.

Second, even if the IP traffic for the IP Sec
connection could be translated, it would fail because the IP
Sec connection is based on Security Associations which
contain the two host IP addresses. These are fundamental to
20 the Security Association architecture (see RFC 2401, *supra*),

in that the inbound IP Sec, on the host where decrypting (or authentication) is to occur, must be uniquely determined by the triple:

{destination IP addr, SPI, IP Sec protocol}.

5 where SPI is the security protocol index (see, RFC 2401, supra).

For example, given hosts A & W, assume NAT is applied to an IP datagram (a generic term for bytes that go on the wire) with ESP in transport mode that is going from A to W.
10 Hence the IP source address is changed. Upon arrival at W, the packet will probably be decrypted successfully since that doesn't depend on IP source address (which was in plaintext -- not tunneled). If strictly implemented however, the inbound SPD checking which should follow
15 decrypting will fail, due to the changed IP source address (because it was not the address used to negotiate the security association). So, even the transport mode ESP case fails.

Simply making NAT and IP Sec mutually exclusive is not
the solution sought by the art. NAT is being deployed
widely because it solves many problems, such as: masks
global address changes, lowers address utilization, lowers
5 Internet service provider (ISP) support burden, and allows
load sharing as virtual hosts.

Yet, NAT is viewed as the greatest single threat to
security integration being deployed in the Internet today.
This "NAT problem", as it is invariably termed, is
10 architecturally fundamental. Yet, legacy applications and
services (for example, those developed for IP version 4)
will continue to a long co-existence as applications and
services develop for IP version 6. Consequently, there is
a great need in the art for providing NAT and IP Sec
15 coexistence, at least in selected situations, and to do so
without introducing serious configuration problems. (IP
version 4 is described in "Internet Protocol", RFC791,
September 1981. IP version 6 is described in Deering, S.,
Hinden, R., "Internet Protocol, Version 6 (IPv6)
20 Specification", RFC2460, December 1998.)

A VPN connection between two address domains can have the effect of directly connecting two domains which most likely will not have been planned to be connected. Hence increased use of VPNs is likely to increase address conflicts. It is also understood that VPNs redefine network visibility and increase the likelihood of address collision when traversing NATs. Address management in the hidden space behind NATs will become a significant burden. There is, therefore, a need in the art to ameliorate that burden.

In U.S. Patent Application Serial No. 09/240,720, a solution to the general problem of integrating IP Sec and NAT is presented. IP security is provided in a virtual private network using network address translation (NAT) by performing one or a combination of the four types of VPN NAT. (Three types of VPN NAT will be further described hereafter, and the fourth is described in copending patent application, assignee docket END9 1999 0093, supra.) This involves dynamically generating NAT rules and associating them with the manual or dynamically generated Internet key exchange (IKE) Security Associations, before beginning IP security that uses the Security Associations. (See, Harkins, D., Carrel, D., "The Internet Key Exchange (IKE)", RFC2409, November 1998. Security Associations is a term

defined in RFC201, supra.) Then, as IP Sec is performed on
outbound and inbound datagrams, the NAT function is also
performed. By "perform IP Sec", is meant to execute the
steps that comprise IP Sec outbound or inbound processing,
5 as defined by the 3 IP Sec RFCs (and others) above. By
"perform NAT", is meant to execute the steps that comprise
the VPN NAT processing hereafter described in this
application.

In U.S. Patent Application Serial No. 09/240,720, the
10 customer must configure each separate VPN NAT rule as a
separate VPN connection. This is time consuming and prone
to error, and VPN connections are really meant to protect
the traffic and should be independent of specific VPN NAT
rules. That is, the rules have heretofore been one to one -
15 NAT thus increases the number of VPN connections required.

It is an object of the invention to provide an improved
and greatly simplified system and method for concurrently
implementing both Network Address Translation (NAT) and IP
Security (IP Sec).

It is a further object of the invention to provide a system and method for solving the increased likelihood of IP address conflicts inherent in the use of a virtual private network (VPN) .

5 It is a further object of the invention to provide a system and method for enabling utilization of VPNs without requiring re-addressing a domain (an expensive alternative) .

10 It is a further object of the invention to provide a system and method for VPN NAT which is accomplished entirely in the IP Sec gateway without requiring changes in domain hosts.

15 It is a further object of the invention to provide a system and method for VPN NAT which requires no, or only minor, changes to routing in each connected domain.

It is a further object of the invention to provide a system and method for VPN NAT which is simple to configure.

It is a further object of the invention to provide a solution to the address collision problems caused by VPNs.

It is a further object of the invention to provide a simplified solution for customer configuration of VPN connections.

It is a further object of the invention to allow a
5 single VPN connection to support multiple VPN NAT rules.

It is a further object of the invention to provide a system and method which, on a system wide basis, avoids conflict among the implicitly, or dynamically assigned, VPN NAT rules.

10 It is a further object of the invention to provide a system and method which reduces system overhead for dynamic NAT rules by eliminating the need to manage numerous separate VPN connections for each NAT rule.

15 It is a further object of the invention to provide a VPN NAT system and method which simplifies network monitoring and traffic analysis.

Summary of the Invention

In accordance with the invention, IP security is provided in a virtual private network using network address translation (NAT) by performing one or a combination of types of VPN NAT. This involves dynamically generating a set of NAT rules and associating the set with the manual or dynamically generated (IKE) Security Associations, before beginning IP security that uses the Security Associations.

Just before IP Sec is performed on outbound datagrams, a specific rule is chosen from the set of NAT rules associated with a connection, and the NAT function is performed, followed by IP Sec. Just after IP Sec is performed on inbound datagrams, a specific rule is chosen from the set of NAT rules associated with a connection, and the NAT function is performed.

Other features and advantages of this invention will become apparent from the following detailed description of the presently preferred embodiment of the invention, taken in conjunction with the accompanying drawings.

Brief Description of the Drawings

Figure 1 is a flow diagram of the VPN NAT method of the preferred embodiment of the invention.

5 Figure 2 illustrates typical IP Sec scenarios and associated VPN NAT pools.

Figure 3 illustrates static NAT, the simplest conventional NAT, for context.

Figure 4 illustrates masquerade NAT, a type of conventional NAT, for context.

10 Figure 5 illustrates VPN NAT, type a (aka 'source-out'): IDci translated for initiator-mode conversations.

Figure 6 illustrates VPN NAT, type c (aka 'source-in'): IDci translated for responder-mode conversations.

15 Figure 7 illustrates VPN NAT, type d (aka 'destination-in'): IDcr translated for responder-mode conversations.

Figure 8 is a high level flow diagram illustrated the relationships between various program and data elements implementing the system and method of the invention.

(Type b relates to 'destination-out' VPN NAT.)

5

Best Mode for Carrying Out the Invention

In accordance with the preferred embodiment of the invention, the NAT problem is addressed through functions VPN NAT with multiple NAT translation rules applied to a single VPN connection, rather than a single NAT translation rule. This greatly simplifies customer configuration. Previously, each address requiring NAT and VPN had to be configured separately. In accordance with the present invention, with reference U.S. Patent Application Serial No. 10/09/240,720, additional function is provided for 'type a source-out' and 'type d destination-in' VPN NAT. To avoid dysfunctional IP Sec connections with the accidental use of HIDE and MAP NAT rules (aka conventional NAT), AH or ESP is checked for during conventional NAT. HIDE and MAP NAT rules are two basic forms of conventional NAT described hereafter in connection with Figures 3 and 4. If a given NAT rule would apply to the IP packet, except for the AH or ESP

header, address translation is not done. This applies to
inbound and outbound NAT. So, the effect is that for
conventional NAT (versus VPN NAT for IP Sec, or IP Sec NAT),
preference is given to IP Sec. IP Sec overrides
conventional NAT.

5

10

15

20

Since it is not known at the time the NAT rules are
loaded whether or not any IP Sec connections might conflict
(dynamic IP for example), checking for such problems cannot
be done until actual NAT processing in the operating system
kernel. User visibility to these actions is provided, if
journaling is on for the rule, by indicating in a journal
entry that a NAT rule fits the datagram, but was not done
due to IP Sec. In addition, operating system kernel
information logging of these actions may be provided for
some limited number of occurrences per conventional NAT
rule. Similarly, a message per connection, rather than per
occurrence, may be provided in a connection manager job log
or in a connection journal. "Journaling" and "journal
entry" are terms also referring to what is typically known
in the art as "logging" and "log entry", respectively.

Pursuant to the invention described in the parent application, referred to as VPN NAT, to allow NAT to be used with IP Sec at the IP Sec gateway, customers retain private internal IP addresses, and increased address collision is avoided by having IP Sec connections begin and end at the IP Sec gateway. An IP Sec gateway is a term defined in RFC2401, *supra*. The term "VPN connection" is another term referring to what is generally called an "IP Sec tunnel", the latter being defined in RFC2401, *supra*.

Further in accordance with the parent application, virtual private networks (VPN) are provided in both initiator and responder modes with an integrated NAT function. Security associations are negotiated using the proper external (NAT rhs) IP addresses, and the NATing of corresponding internal (NAT lhs) IP addresses is done by generated NAT rules, in sync with connection load to IPsec and IPsec processing in Operating system kernel. Inbound source IP addresses are translated, as well as the usual source IP address NAT on outbound (with corresponding translation of destination IP address on inbound). A 'VPN NAT rule' is represented by blocks 126, 124 in Figure 5; that is, the 2 sets of lhs and rhs addresses comprise a VPN NAT rule.

In accordance with the present invention, a single VPN connection supports multiple VPN NAT rules by allowing the customer to specify a class of NAT rules associated with a VPN connection, and allow the system to dynamically generated a specific NAT binding (rule) out of this class.

5 Further, on a system wide basis, conflict among the implicitly, or dynamically assigned, VPN NAT rules is avoided by associating customer configured NAT address pools with local IP addresses when the VPN NAT type is source

10 inbound. The three types of VPN NAT of concern to this application are defined by Table 1, infra.

Referring to Figure 1, the method of the preferred embodiment of the invention for executing VPN NAT includes in step 20 configuring connections that require NAT, in step 15 22 defining IPSec NAT address pools, in step 24 starting initiator mode connections, in step 26 starting responder mode connections (these are started at the other end of the connection), in step 28 processing SA pair updates, and in step 30 ending the connections. (A NAT pool is a set of IP 20 addresses.) Each of these steps is further explained below.

In step 20, the user decides on and configures the connections that will require NAT. This is logically equivalent to writing NAT rules. The four cases to be considered in doing so are depicted in Table 1.

5

TABLE 1: TYPES OF VPN NAT

		IDci (source)	IDcr (destination)
10	Initiator Mode	source-out type a.NAT internal address, IP src on outbound, IP dest on inbound.	destination-out type b.NAT
15	Responder Mode	source-in type c.NAT external address, IP src on inbound, IP dest on outbound.	destination-in type d.NAT internal address, IP dest on inbound, IP src on outbound.
20	where	IDci = 'identifier of client initiator', IDcr = 'identifier of client responder'.	

A VPN connection is defined as having four endpoints: two 'connection endpoints', and two 'data endpoints'.
25 (Transport mode then means that the connection endpoint equals the data endpoint, at each end of the connection.)
The IDci and IDcr terms refer to the two data endpoints, more specifically, by indicating which is the initiator and which is the responder (see, RFC2409, supra.) Also, these

identifiers may take one of about six different forms, which are part of the IDcr, IDcr definitions. For this application, identifier types are not particularly relevant.

When specifying a specific instance of NAT in, for
5 example, an IP Sec Policy database, the user makes a yes/no decision in, say, a check-box. As used herein, an IP Sec policy refers to the complete set of configured IP Sec information, on a system. This information is stored in what is termed the IP Sec database, or IP Sec policy
10 database. Responder mode NAT flags IDci and IDcr may be part of the connection definition. The initiator mode flag may be part of the user client pair, associated with a 'local client ID' (only). The responder IDci and IDcr NAT flags can be set independently. Both are relevant only if
15 the connection definition has external initialization mode.

Heretofore, in all cases, if the NAT flag was 'on', the corresponding granularity value was required to be 's' (scalar) in the connection definition. In accordance with the present invention, this is no longer a restriction with
20 dynamic VPN NAT. That is, granularity of 's' (scalar), 'f' (filter) and 'c' (client) are all supported. 'Granularity' is described in RFC2401, supra, at pages 15-16. In

accordance with an exemplary embodiment of the invention,
for example the IBM AS/400, 'granularity' is implemented as
follows: each VPN connection has five selectors (fields in
datagram that might be checked to determine if traffic
5 should be in the VPN connection; these are: source IP, dest
IP, source port, destination port and protocol. In
accordance with this exemplary embodiment, when a VPN
connection is started, each selector get its value from
either (1) the policy filter for that VPN connection (for
10 selector granularity 'f'), (2) single values from IKE (for
selector granularity 's'), or (3) contiguous range of values
from IKE (for selector granularity 'c').

Referring to Figure 2, the manner in which VPN NAT IP
pools relate to network scenarios is shown. Lines 34 and 36
15 represent IP Sec connections between gateways (GW) 42, 44
and 46 on Internet 40. NAT pools 52, 54 for types 'a
source-out' and 'c source-in' are independently associated
with each remote ID (gateway 44, 46). For type 'd
destination-in' VPN NAT, a single pool 50 may be defined for
20 global IP addresses that the VPN NAT gateway 42 owns. In
this exemplary embodiment, IP SEC policies for NAT pools 50,
52, 54 are stored in IP SEC data base 48. In this example,
all three internal networks 56, 58 and 60 use the same

10.*.*.* addresses space. This provides the initial value
and motivation for VPN NAT: IP Sec tunnels (aka connections)
between these internal networks 56, 58, 60 has a logical
effect combining them. This cannot be done, in general,
5 without address conflict. VPN NAT provides the solution to
the problem presented to gateway (Gw 1) 42 when it needs to
do business with hosts behind gateways (Gw Q) 44 and (Gw Y)
46 on internal networks 60 and 58, respectively.

In step 22, the user defines a set (in pools 50, 52 and
10 54) of IP addresses that are available for the exclusive use
of the VPN NAT function. Each pool is preferably definable
as a range of IP address, but could be a list of
discontiguous addresses, and is naturally associated with
remote ID and local ID IP Sec Policy database entities.

15 Referring to Table 2, the different meanings of each
flavor of VPN NAT motivating the different pools are set
forth. Although specified on a per remote ID or local ID
basis, the pools may be managed as three distinct groups of
IP addresses. This allows the user to specify, for example,
20 the same range for multiple remote ID's. The letters a, c

and d correspond to the VPN NAT types (Table 1). The column 'lr?' means locally routable (as distinguished from globally routable.)

5

TABLE 2: IP SEC NAT POOLS

	IP Sec NAT pools	Pool purpose	Effective generated outbound NAT rule	lr?
10	a. Locally initiated connections	1. Hide own IP translate addresses from remote hosts	"MAP srcIP TO <value from pool>".	yes
15		2. Avoid IP address conflict with remote GW	Hence, a pool may be associated with each remote ID.	
20				
25	c. Remotely initiated connections	Avoid address conflict	"map destIP to <value from pool>"	yes
30	d. Remotely initiated connections	1. Provide a form of load sharing	"map srcIP to <value from pool>"	yes
35		2. Hide own IP addresses behind external address.		

Hence, a pool
may be
associated with
a globally
routable IP
address (IDcr).

5

In step 24, initiator mode connections are started.
When starting an initiator mode connection, the connection
10 manager checks if the local client ID is to be translated.
If so, the connection manager looks for an available IP
address from NAT pool, say 52, associated with a remote ID
in the database. Availability is determined by the
connection manager as follows. The connection manager is a
15 server which, running all of the time, starts and stops VPN
connections and provides status. This server maintains a
single (system-wide, since connection manager runs once per
system) list of IP addresses that have been used in some
active connection (states: starting, running or stopped)
20 from any type 'a source-out' pool (see Table 1). The first
IP address in the pool not in the used list, is chosen, and
added to the used list. If an available IP address cannot
be found, the connection is not started and an appropriate
error message (and possibly return code to the OP NAV GUI)
25 is generated. The policy database is not updated to show an
IP address is in use -- rather this is determined
dynamically by the connection manager based solely on its

set of active connections. An 'OP NAV GUI' is an "AS/400 Operations Navigator graphical user interface (GUI)", a PC-based GUI used to configure various aspects of AS/400, including VPN.

5 The start message (msg) sent by connection manager to
IKE will have the NAT rhs IP address selected from the pool.
The NAT rhs IP address is added to the security association
(SA) pair, which is completed by the returned SAs from IKE.
Connection manager then loads the connection to IPSec. An
10 SA pair is two security associations (defined by RFC2401,
supra), one inbound and one outbound.

IPSec generates NAT rules for the two SAs. On
outbound, NAT will occur after filtering and before IPSec
and on inbound, NAT will occur after IPSec (and before
15 filtering, if any). In this sense, NAT is 'wrapping' the
local connection endpoint of the IPSec connection.

Referring to Figures 3 and 4, conventional NAT
functions are illustrated for background and contrast with
later figures which show VPN NAT types according to the
20 invention.

Referring to Figure 3, static is the simplest form of NAT. Both conventional NAT types are explicitly configured by the user by writing the corresponding NAT rule statements via the OpNav GUI. This is in contrast to the IPSec NAT, in which the actual NAT rules or statements are generated by the system. The MAP statement <MAP lhs TO rhs> of Figure 3 and the HIDE statement <HIDE ip addr set BEHIND rhs> of Figure 4 are such statements.

Again referring to Figure 3, on inbound processing, if source ip 70 matches lhs 72 in the MAP lhs TO rhs statement, then src ip 70 is translated to rhs 76. On outbound processing, if destination ip 74 matches rhs 76, then destination ip 74 is translated to lhs 72.

Referring to Figure 4, masquerade NAT (also referred to as network address and port translation (NAPT)), uses the HIDE statement, supra, and provides many-to-one address translation by using its own port pools 118 (UDP, TCP) to remember how to translate the inbound traffic. Unlike static NAT (Figure 3), masquerade NAT conversations <CONVERSATION src ip, src port, rhs ip, rhs port,...> can only be initiated by internal (lhs) addresses. VPN NAT, a name used to identify the preferred embodiment of the

present invention, as will be seen, is closer to static NAT, in that it does not include port translation.

Referring further to Figure 4, in processing outbound datagrams, in step <1> if source ip address 90 is determined
5 to be in the ip address set 92 of the HIDE statement, then in step <2> the CONVERSATION is set up by copying src ip 90 into CONVERSATION field 94, in step <3> source port 98 into field 96, in step <4> rhs 104 into field 100, and in step <5> the rhs port into field 102 from the correct pool in
10 port pools 118. Then, in step <6> source ip 90 is translated to rhs 104, and in step <7> source port 98 is changed to rhs port 102. In processing inbound datagrams, if in step <8> destination ip address 106 and destination port 108 match CONVERSATION fields rhs ip 100 and rhs port
15 102, respectively, then in step <9> destination ip address 106 is translated to CONVERSATION source ip address 94 and in step <10> destination port 108 is translated to CONVERSATION source port 96.

Some special situations also handled by NAT are not
20 illustrated because they are of no interest to the present invention. These include handling of special situations created by FTP or ICMP, both of which contain IP address

that are translated. FTP = File Transfer Protocol (defined
in RFC959), and ICMP = Internet Control Message Protocol
(defined in RFC792). Checksum re-calculation is done. In
masquerade NAT once a conversation exists, later datagrams
5 are matched against that, rather than the original
(precipitating) HIDE rule, the port pools are managed,
conversations are timed and terminated, and ports are
mapped. It is a particular advantage of the invention that
VPN NAT supports ICMP and FTP (including the famous FTP PORT
10 and PASV commands and attendant problems).

In accordance with the present invention, dynamically
determined VPN NAT rules are implemented as follows. The
customer specifies, via a graphical user interface (GUI)
that VPN NATing is to be done. Multiple IP addresses are
15 allowed for the source IP address of locally initiated
connections. These multiple IP addresses are specified via
range (contiguous) or address and mask. These constitute
the VPN NAT rule left-hand-side (lhs) address set. The VPN
NAT rule right-hand-side (rhs) address set is associated
20 with the remote VPN gateway address. When a connection is
started, both lhs and rhs address sets are loaded with the
connection as part of the IP Sec Security associations for
the connection. The VPN Connection manager then marks the

rhs set as used to avoid NAT rule conflict, with connections started later.

As IP traffic occurs for a loaded (that is, installed into the operating system kernel), locally initiated, 5 connection, the lhs and rhs address sets are used to determine what specific NAT rule should be applied to a particular datagram. The generation of a datagram-specific NAT rule is done by ordering each address set and each address in the lhs set is mapped, one-to-one, with the 10 corresponding element of the rhs set. If the lhs set cardinality is larger than the rhs set, VPN NAT will not occur for the n elements of the lhs set where $n >$ cardinality(rhs). (However, this may be prevented by audit at GUI level.) For outbound traffic, the n'th element of 15 the lhs set is selected based on the datagram source IP address, and for inbound traffic the n'th element of the rhs set is selected based on (that is, equal to) the datagram destination IP address.

As IP traffic occurs for a loaded, remotely initiated 20 connection (responder mode), the solution is essentially the same as for locally initiated connections, but reversed. In this case, the rhs set is matched against the inbound

datagram destination IP address, which is mapped to the corresponding element of the lhs set. If the rhs set is greater than the lhs, NATing would simply not occur. Again, this may be undesirable from a human factor perspective, in
5 which case it may be disallowed by audit at the GUI level.

Referring to Figures 5, 6 and 7, lhs and rhs refer to sets, such as contiguous ranges, of IP addresses. Assuming that x is a set, then $\text{size}(x)$ designates the number of elements in the set. Three cases are provided, as follows:

10 Case 1: $\text{size}(\text{lhs})=\text{size}(\text{rhs})=1$.

Case 2: $\text{size}(\text{lhs})=\text{size}(\text{rhs}) \& \text{size}(\text{lhs}) > 1$.

Case 3: $\text{size}(\text{lhs}) \neq \text{size}(\text{rhs})$.

Case 1 is handled by the system and method of the parent application, U.S. Patent Application Serial No.

15 09/240,720.

In case 2, since the two sets are equal, the implicit MAP rule generated for each connection as it is started is inherent in the statement of the two sets. That is, there

is a unique one-to-one correspondence between elements of
the lhs and elements of the rhs. So, the generation of the
implicit MAP rule for a particular VPN connection load is
straightforward. For the process of Figure 5, for example,
5 for source out VPN NAT, the n'th element of the lhs set that
matches the source IP (step 1) is found, then the n'th
element of the rhs set is found and used to replace the
source IP (step 2).

In case 3, a dynamic association (a binding) of the lhs
10 element with the rhs element is generated based on
previously generated bindings. A binding is generated as
needed, by traffic, or an existing binding is used. A
binding lasts for the duration of the connection or until an
inactivity time-out value is reached. The bindings are of
15 two types: local and remote, and are unique across the
system.

Which case is determined once, per VPN connection, at
the time it is started, and not recomputed for each datagram
handled.

20 Referring to Figure 5, the preferred embodiment of the
invention for VPN NAT type a 'source-out' is illustrated.

In VPN NAT, type a 'source-out', IDci is translated for initiator-mode conversations. After system generated implicit NAT rule 128 <MAP lhs TO rhs> is loaded, it functions as static NAT. The key to making this work, is
5 that the security associations negotiated by IKE use the implicit MAP 130 rhs 138. Hence, the SAs and the VPN NAT are synchronized.

Referring further to Figure 5, for a locally initiated conversation, in step <-2>, since NAT is requested, implicit
10 MAP rule 128 is created by copying local client ID 122 to lhs 126, and the rhs 124 is obtained from the appropriate pool 120. Step <0> is part of starting a VPN connection, and occurs during steps 24 and 26 (Figure 1). In step <0>, after IKE negotiation is complete using rhs 124, implicit
15 MAP rule 130 is loaded to the operating system kernel. This step <0> comprises the following steps; load the connection SA's, connection filter, and create blank version of table 210. For outbound processing, if in step <1> src ip 132 matches any particular lhs in implicit map rule 130, then in
20 step <2> case 1, 2 or 3 (described above) is determined, resulting in a rhs 138 IP address. This selected rhs replaces source IP 132. An entry of the selected binding is made in the local binding table 210, if case 3. For inbound

processing, if in step <3> dest ip address 140 matches a rhs in the local binding table 210, then in step <4> destination ip 140 is replaced by the lhs of the local binding table entry 210.

5 lhs and rhs are two sets of IP addresses. A VPN NAT rule consists of one each, that is, it defines a mapping of lhs addresses on rhs addresses: lhs -> rhs.

10 In step 26 (Figure 1)+, responder mode connections are started. In so doing, IKE functions negotiate the SAs based on currently configured policy. When done, they are sent to the connection manager as a SA collection of 1 to n SA pairs.

15 In Figures 6 and 7, VPN NAT source-in and destination-in types are illustrated.

20 Referring to Figure 8, the connection manager server 300, upon receiving the start message (msg) 332 from IKE server 330, looks at the connection definition 306 in the database 304 and checks the NAT flags 314. If one or more NAT remote flags so 308, si 310, or di 312 is 'on', then an IP address(es) 154 (Figure 6), 186 (Figure 7) is obtained

from the appropriate NAT pool 50, 52 or 54 (Figure 2), depending on NAT flag associated with the ID 152 in (Figure 6), 182 (Figure 7.)

The relationship between the NAT flag and NAT pools is;
5 if source-out flag 308 is on or source-in flag 310 is on, pool type 52 (same type as 54) in Figure 2 is used, selecting the pool based on remote VPN connection endpoint address. If destination-in flag 312 is on, pool 50 in Figure 2 is used, indexed by destination address.

10

Management of IP address availability from the remote ID pool 150 is done by the connection manager based on its set of active connections (as for type a 'source-out' VPN NAT). Connection manager also handles availability for the 15 IDcr pool 180 (Figure 7), which allows load balancing. The IDcr pool 180 is a set of IP addresses for nat'ing IDcr. There are two basic approaches: (1) for every start search the pool 180 from the first entry; or, (2) for every start, the pool 180 is searched from the last used IP.

The load to IPSec occurs as in the initiator mode case above. When processing remotely initiated connection traffic, two address translations may occur for each inbound and outbound packet (source and destination).

5 Referring to Figure 6, VPN NAT type c 'source-in' starts a responder-mode connection as follows: in step <-2>, implicit MAP rule 158 <MAP lhs TO rhs> is created, by copying IDci 152 to rhs 154; and in step <-1>, by selecting ip address(es) from the appropriate pool 150 and copied to
10 lhs 156. In step <0>, after IKE negotiation is complete using rhs 154, implicit rule 160 is loaded. This step <0> includes the following: same as above -- step <0> is the same in all three VPN NAT types (except for some low-level details). When processing inbound datagrams, if in step <1>
15 src ip 172 matches a rhs 168, in step <2>, source ip 172 is translated to corresponding lhs 166. Then, based on case 1,2 or 3 (as described above), an entry is made in the remote binding table 212. When processing outbound datagrams, if in step <3> destination IP 164 matches lhs
20 166, then in step <4> destination ip 164 is translated to rhs 168. The lookup for IP 164 used the remote binding table, if case 3, else it uses the implicit MAP rule 160.

Referring to Figure 7, VPN NAT d 'destination-in' type executes to translate IDcr for responder-mode conversations as follows: in step <-2> implicit MAP rule 188 is created, copying IDcr 182 to rhs 184. In step <-1>, ip address(es) are obtained from appropriate address pool 180 and copied to lhs 186. In step <0>, after IKE negotiations are completed using rhs 184, implicit MAP rule 190 is loaded. (Step <0> is the same as for Figures 5 and 6, except for low-level details.)

When processing inbound datagrams if in step <1> destination ip 200 matches rhs 198, in step <2> destination ip 200 is translated to lhs 196. When processing outbound datagrams if in step <3> source ip 192 matches lhs 196, in step <4> source ip 192 is translated to rhs 198.

Referring to Figure 8, in step 28, when the connection manager 300 gets SA pair updates 302, it copies the new SA pair information to the SA pair table 322 in connection process memory 320.

In step 30, when ending a connection 34, 36, the connection manager 300 frees (makes available) any NAT IP addresses 52, 54 associated with the connection. Referring

to Figure 8, NAT IP addresses are removed from the appropriate list 316 maintained by the connection manager 300.

The size of the lhs and rhs sets is controlled by
5 taking the minimum of three items: the subnet size (or address range) configured by the customer, the maximum concurrent VPN NAT sessions per connection configured by the customer on a per NAT pool basis, and the size of the largest remaining range of value still available in the
10 originally configured pool. This is determined by the VPN connection manager during the startup of a connection (step 24 and 26, Figure 1).

Advantages over the Prior Art

It is an advantage of the invention that there is
15 provided an improved and greatly simplified system and method for concurrently implementing both Network Address Translation (NAT) and IP Security (IP Sec).

It is a further advantage of the invention that there is provided a system and method for solving the increased likelihood of IP address conflicts inherent in the use of a virtual private network (VPN).

5 It is a further advantage of the invention that there is provided a system and method for enabling utilization of VPNs without requiring re-addressing a domain (an expensive alternative).

10 It is a further advantage of the invention that there is provided a system and method for VPN NAT which is accomplished entirely in the IP Sec gateway without requiring changes in domain hosts.

15 It is a further advantage of the invention that there is provided a system and method for VPN NAT which requires no, or only minor, changes to routing in each connected domain.

It is a further advantage of the invention that there is provided a system and method for VPN NAT which is simple to configure.

It is a further advantage of the invention that there is provided a solution to the address collision problems engendered by VPNs.

5 It is a further advantage of the invention that there is provided a simplified solution for customer configuration of VPN connections.

It is a further advantage of the invention that there is provided a system and method allowing a single VPN connection to support multiple VPN NAT rules.

10 It is a further advantage of the invention that there is provided a system and method which, on a system wide basis, avoids conflict among the implicitly, or dynamically assigned, VPN NAT rules.

15 It is a further advantage of the invention that there is provided a system and method which reduces system overhead in systems dynamic NAT rules by eliminating the need to manage numerous separate VPN connections for each NAT rule.

It is a further advantage of the invention that there is provided a VPN NAT system and method which simplifies network monitoring and traffic analysis.

5 It is a further advantage of the invention that there is provided a VPN NAT system and method which simplifies network monitoring and traffic analysis.

Alternative Embodiments

It will be appreciated that, although specific 10 embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. In particular, it is within the scope of the invention to provide a computer program product or program 15 element, or a program storage or memory device such as a solid or fluid transmission medium, magnetic or optical wire, tape or disc, or the like, for storing signals readable by a machine, for controlling the operation of a computer according to the method of the invention and/or to 20 structure its components in accordance with the system of the invention.

Further, each step of the method may be executed on any general computer, such as an IBM System 390, AS/400, PC or the like and pursuant to one or more, or a part of one or more, program elements, modules or objects generated from any programming language, such as C++, Java, Pl/I, Fortran or the like. And still further, each said step, or a file or object or the like implementing each said step, may be executed by special purpose hardware or a circuit module designed for that purpose.

10

Accordingly, the scope of protection of this invention is limited only by the following claims and their equivalents.

03528345 - 0532300

CLAIMS

We claim:

1 1. A method of operating a virtual private network (VPN)
2 based on IP Sec that integrates network address
3 translation (NAT) with IP Sec processing, comprising
4 the steps of:

5 configuring a NAT IP address pool;

6 configuring a VPN connection to utilize said NAT IP
7 address pool;

8 obtaining a specific IP address from said NAT IP
9 address pool, and allocating said specific IP address
10 for said VPN connection;

11 starting said VPN connection;

12 loading to an operating system kernel the security
13 associations and connection filters for said VPN
14 connection;

15 processing a IP datagram for said VPN connection; and

16 applying VPN NAT to said IP datagram.

1 2. The method of claim 1, wherein said VPN connection is
2 configured for outbound processing, and said applying
3 step comprises outbound source IP Nating.

1 3. The method of claim 1, wherein said VPN connection is
2 configured for some combination of inbound processing,
3 and said applying step selectively comprises inbound
4 source IP NATing or inbound destination IP NATing.

1 4. The method of claim 1, further for integration of NAT
2 with IP Sec for manually-keyed IP Sec connections,
3 comprising the further step of manually configuring
4 connection keys.

1 5. The method of claim 1, further for integrating NAT with
2 IP sec for dynamically-keyed (e.g. IKE) IP Sec
3 connections, comprising the further step of:
4 configuring the VPN connections to obtain their keys
5 automatically.

1 6. The method of claim 1, further for integrating NAT with
2 IP Sec Security Associations, negotiated dynamically by
3 IKE, wherein said starting step further comprises
4 creating a message for IKE containing said IP address
5 from said NAT pool; and further comprising the step of
6 operating IKE to obtain dynamically negotiated keys.

1 7. The method of claim 6, further comprising the step of
2 combining the dynamically obtained keys with said NAT
3 pool IP address and wherein said loading step loads the
4 result as security associations into said operating
5 system kernel.

1 8. A method for allowing the definition and configuration
2 of NAT directly with definition and configuration of
3 IPsec-based VPN connections and VPN policy, comprising
4 the steps of:

5 configuring the requirement for VPN NAT by a yes/no
6 decision in a policy database for each of the three
7 types of VPN NAT, said three types being VPN NAT type a

8 outbound source IP NAT, VPN NAT type c inbound source
9 IP NAT, and VPN NAT type d inbound destination IP NAT;
10 and

11 configuring a remote IP address pool or a server IP
12 address pool selectively responsive to said yes/no
13 decision for each said VPN NAT type.

1 9. The method of claim 8, further comprising the step of
2 configuring a unique said remote IP address pool for
3 each remote address to which a VPN connection will be
4 required, whereby said remote IP address pool is keyed
5 by a remote ID.

1 10. The method of claim 8, further comprising the step of
2 configuring said server IP address pool once for a
3 system being configured.

1 11. A method of providing customer tracking of VPN NAT
2 activities as they occur in an operating system kernel,
3 comprising the steps of:

4 responsive to VPN connection configuration, generating
5 journal records;

6 updating said journal records with new records for each
7 datagram processed through a VPN connection; and

8 enabling a customer to manage said journal records.

1 12. A method of allowing a VPN NAT address pool to be
2 associated with a gateway, thereby allowing server
3 load-balancing, comprising the steps of:

4 configuring a server NAT IP address pool for a system
5 being configured;

6 storing specific IP addresses that are globally
7 routable in said server NAT IP address pool;

8 configuring a VPN connection to utilize said server NAT
9 IP address pool; and

10 managing total volume of concurrent VPN connections
11 responsive to the number of addresses in said server
12 NAT IP address pool.

1 13. A method of controlling the total number of VPN
2 connections for a system based on availability of NAT
3 addresses, comprising the steps of:

4 configuring the totality of remote IP address pools
5 with a common set of IP addresses, said addresses being
6 configured as a range, as a list of single addresses,
7 or any combination of multiple ranges and single
8 addresses; and

9 limiting the successful start of concurrently active
10 VPN connections responsive to the number of said IP
11 addresses configured across the totality of said remote
12 address pools.

1 14. A method of performing network address translation on
2 selected ICMP datagrams, comprising the steps of:

3 detecting selected types of ICMP type packets; and
4 responsive to said selected types, performing network
5 address translation functions on the entire datagram
6 including ICMP data.

1 15. A method of performing network address translation on
2 selected FTP datagrams, comprising the steps of:

3 detecting the occurrence of FTP PORT or PASV FTP
4 commands; and

5 responsive to said command, performing network address
6 translation on the FTP data and the header.

1 16. A system for operating a virtual private network (VPN)
2 based on IP Sec that integrates network address
3 translation (NAT) with IP Sec processing, comprising:

4 means for configuring a NAT IP address pool;

5 means for configuring a VPN connection to utilize said
6 NAT IP address pool;

7 means for obtaining a specific IP address from said NAT
8 IP address pool, and allocating said specific IP
9 address for said VPN connection;

10 means for starting said VPN connection;

11 means for loading to an operating system kernal the
12 security associations and connection filters for said
13 VPN connection;

14 means for processing a IP datagram for said VPN
15 connection; and

16 means for applying VPN NAT to said IP datagram.

1 17. A system for definition and configuration of NAT
2 directly with definition and configuration of VPN
3 connections and VPN policy, comprising:

4 a policy database for configuring the requirement for
5 VPN NAT by a yes/no decision for each of the three
6 types of VPN NAT, said three types being VPN NAT type a
7 outbound source IP NAT, VPN NAT type c inbound source
8 IP NAT, and VPN NAT type d inbound destination IP NAT;
9 and

10 a remote IP address pool or a server IP address pool
11 selectively configured responsive to said yes/no
12 decision for each said VPN NAT type.

1 18. A system for allowing a VPN NAT address pool to be
2 associated with a gateway, thereby allowing server
3 load- balancing, comprising:

4 a server NAT IP address pool configured for a given
5 system being configured for containing multiple address
6 configured as a range, as a list of single addresses,
7 or any combination multiple ranges and single
8 addresses;

9 said server NAT IP address pool storing specific IP
10 addresses that are globally routable;

11 a VPN connection configured to utilize said server NAT
12 IP address pool; and

13 a connection controller for managing total volume of
14 concurrent VPN connections responsive to the number of
15 addresses in said server NAT address pool.

1 19. A program storage device readable by a machine,
2 tangibly embodying a program of instructions executable
3 by a machine to perform method steps for operating a
4 virtual private network (VPN) based on IP Sec that

5 integrates network address translation (NAT) with IP
6 Sec processing, said method steps comprising:

7 configuring a NAT IP address pool;

8 configuring a VPN connection to utilize said NAT IP
9 address pool;

10 obtaining a specific IP address from said NAT IP
11 address pool, and allocating said specific IP address
12 for said VPN connection;

13 starting said VPN connection;

14 loading to an operating system kernal the security
15 associations and connection filters for said VPN
16 connection;

17 processing a IP datagram for said VPN connection; and

18 applying VPN NAT to said IP datagram.

1 20. An article of manufacture comprising:

2 a computer useable medium having computer readable
3 program code means embodied therein for operating a
4 virtual private network (VPN) based on IP Sec that
5 integrates network address translation (NAT) with IP
6 Sec processing , the computer readable program means in
7 said article of manufacture comprising:

8 computer readable program code means for causing a
9 computer to effect configuring a NAT IP address pool;

10 computer readable program code means for causing a
11 computer to effect configuring a VPN connection to
12 utilize said NAT IP address pool;

13 computer readable program code means for causing a
14 computer to effect obtaining a specific IP address from
15 said NAT IP address pool, and allocating said specific
16 IP address for said VPN connection;

17 computer readable program code means for causing a
18 computer to effect starting said VPN connection;

19 computer readable program code means for causing a
20 computer to effect loading to an operating system
21 kernel the security associations and connection filters
22 for said VPN connection;

23 computer readable program code means for causing a
24 computer to effect processing a IP datagram for said
25 VPN connection; and

26 computer readable program code means for causing a
27 computer to effect applying VPN NAT to said IP
28 datagram.

1 21. Method for providing IP security in a virtual private
2 network using network address translation (NAT),
3 comprising the steps of:

4 dynamically generating NAT rules and associating them
5 with manual or dynamically generated (IKE) Security
6 Associations; thereafter

7 beginning IP security that uses the Security
8 Associations; and then

9 as IP Sec is performed on outbound and inbound
10 datagrams, selectively performing one or more of VPN
11 NAT type a outbound source IP NAT, VPN NAT type c
12 inbound source IP NAT, and VPN NAT type d inbound
13 destination IP NAT.

1 22. The method of claim 1, said NAT IP address pool
2 containing multiple addresses configured as a range, as
3 a list of single address, or any combination of
4 multiple ranges and single addresses.

00000000000000000000000000000000

**SYSTEM AND METHOD FOR NETWORK ADDRESS TRANSLATION
INTEGRATION WITH IP SECURITY**

Abstract of the Disclosure

IP security is provided in a virtual private network
5 using network address translation (NAT) by performing one or
a combination of the four types of VPN NAT, including VPN
NAT type 'a source-outbound' IP NAT, VPN NAT type 'b
destination-outbound, VPN NAT type 'c inbound-source' IP
NAT, and VPN NAT type 'd inbound-destination' IP NAT. This
10 involves dynamically generating NAT rules and associating
them with the manual or dynamically generated (IKE) Security
Associations, before beginning IP security that uses the
Security Associations. Then, as IP Sec is performed on
outbound and inbound datagrams, the NAT function is also
15 performed.

095578245-052300

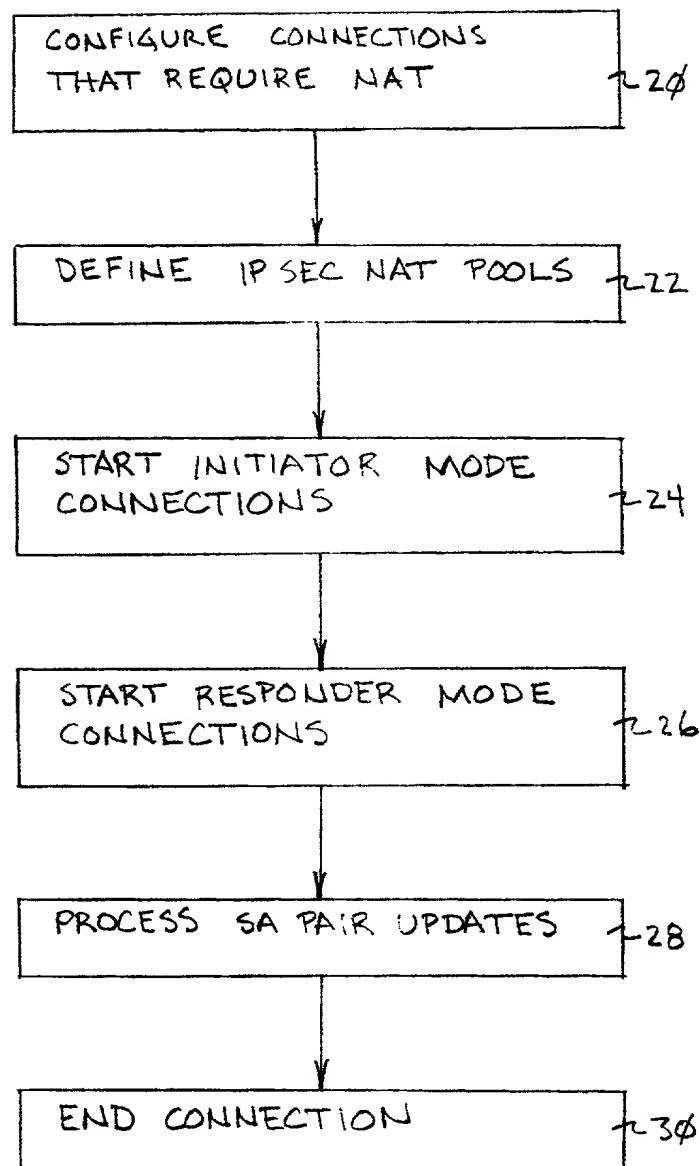


FIG. 1

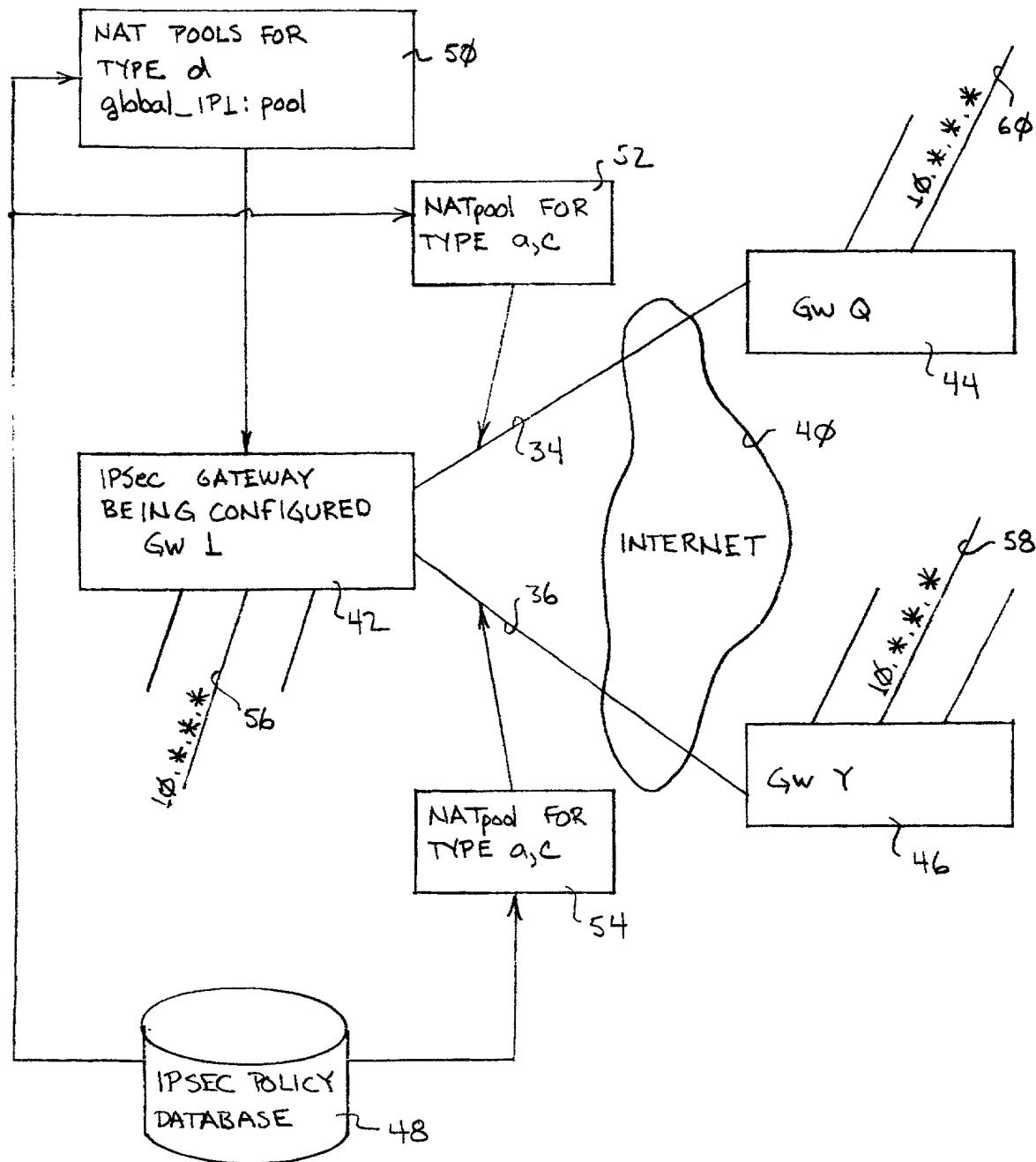


FIG. 2

END9 1999 Ø129 US1
3/8

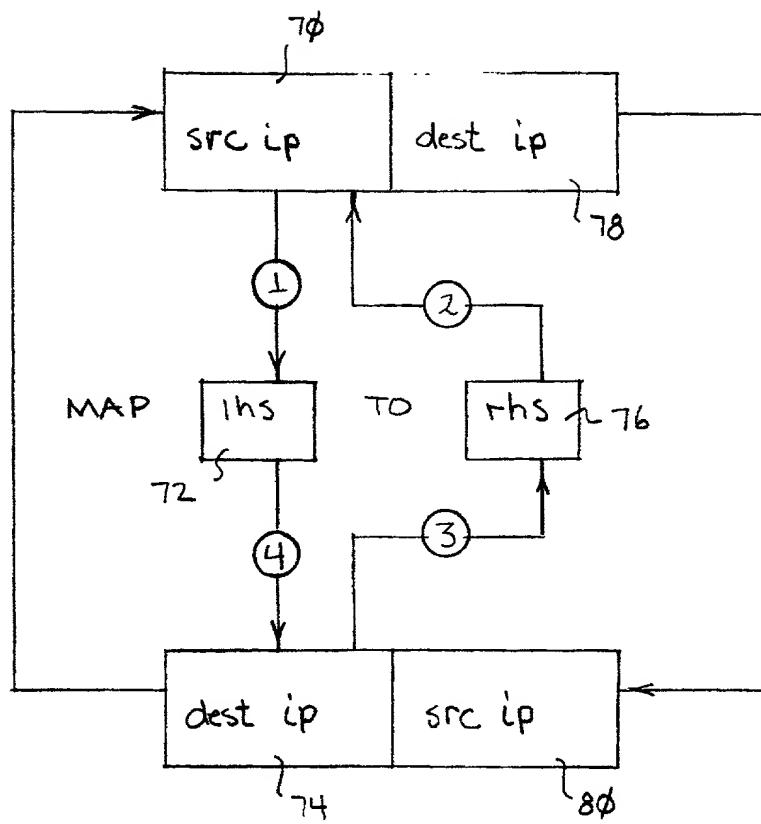


FIG. 3.

END9 L999 φ129 USL
4/8

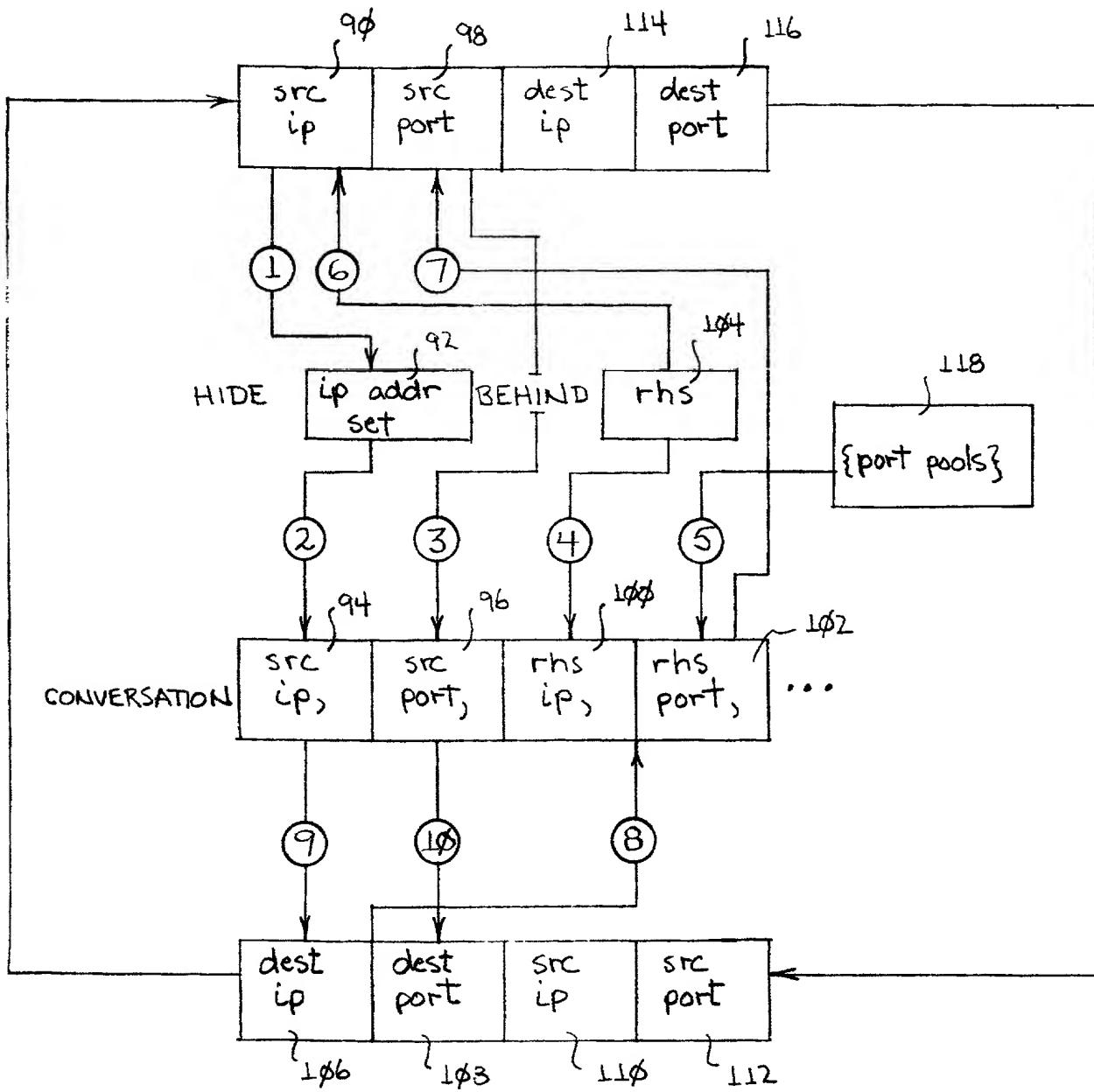


FIG. 4

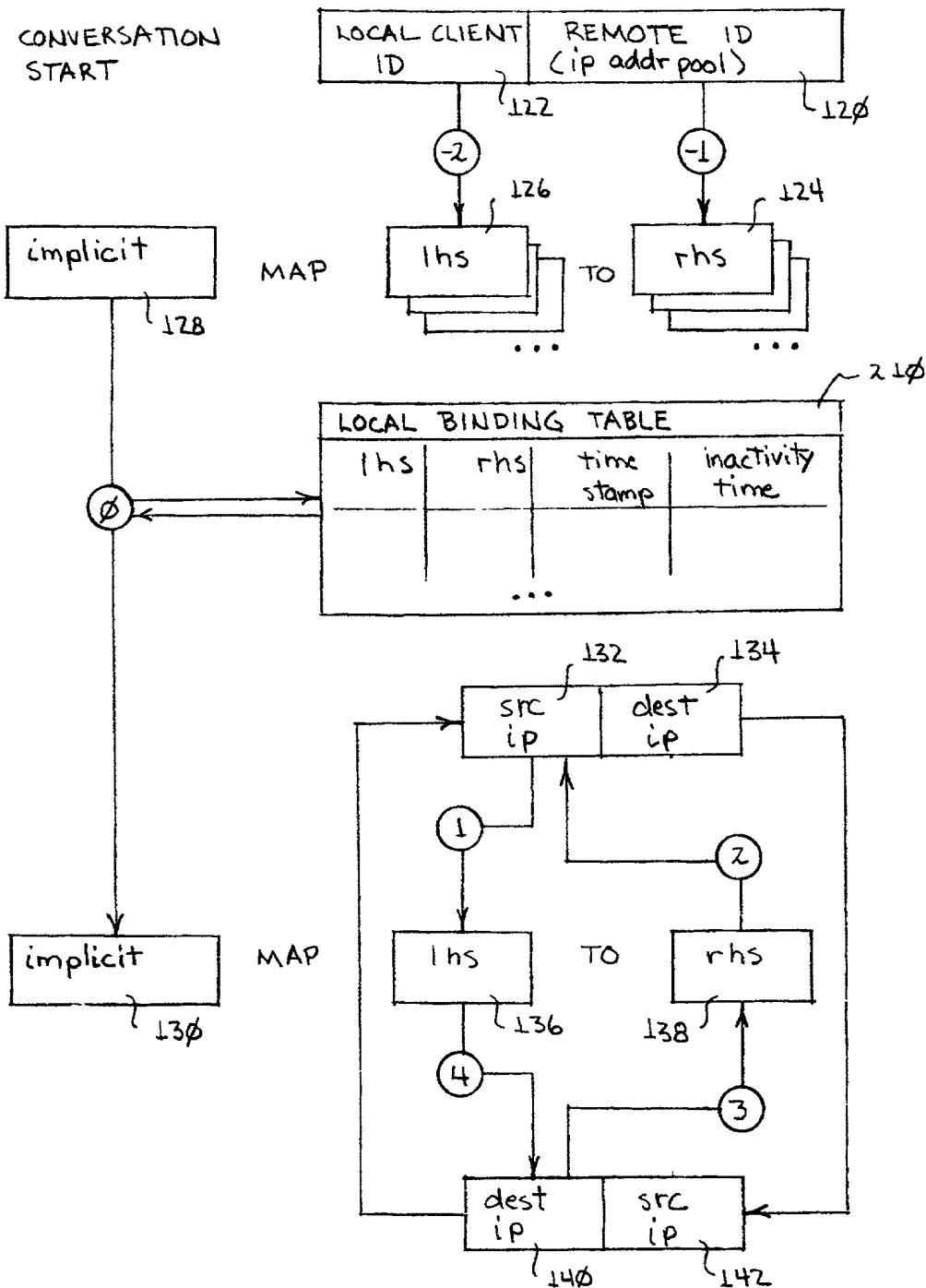


FIG.5

CONVERSATION
START

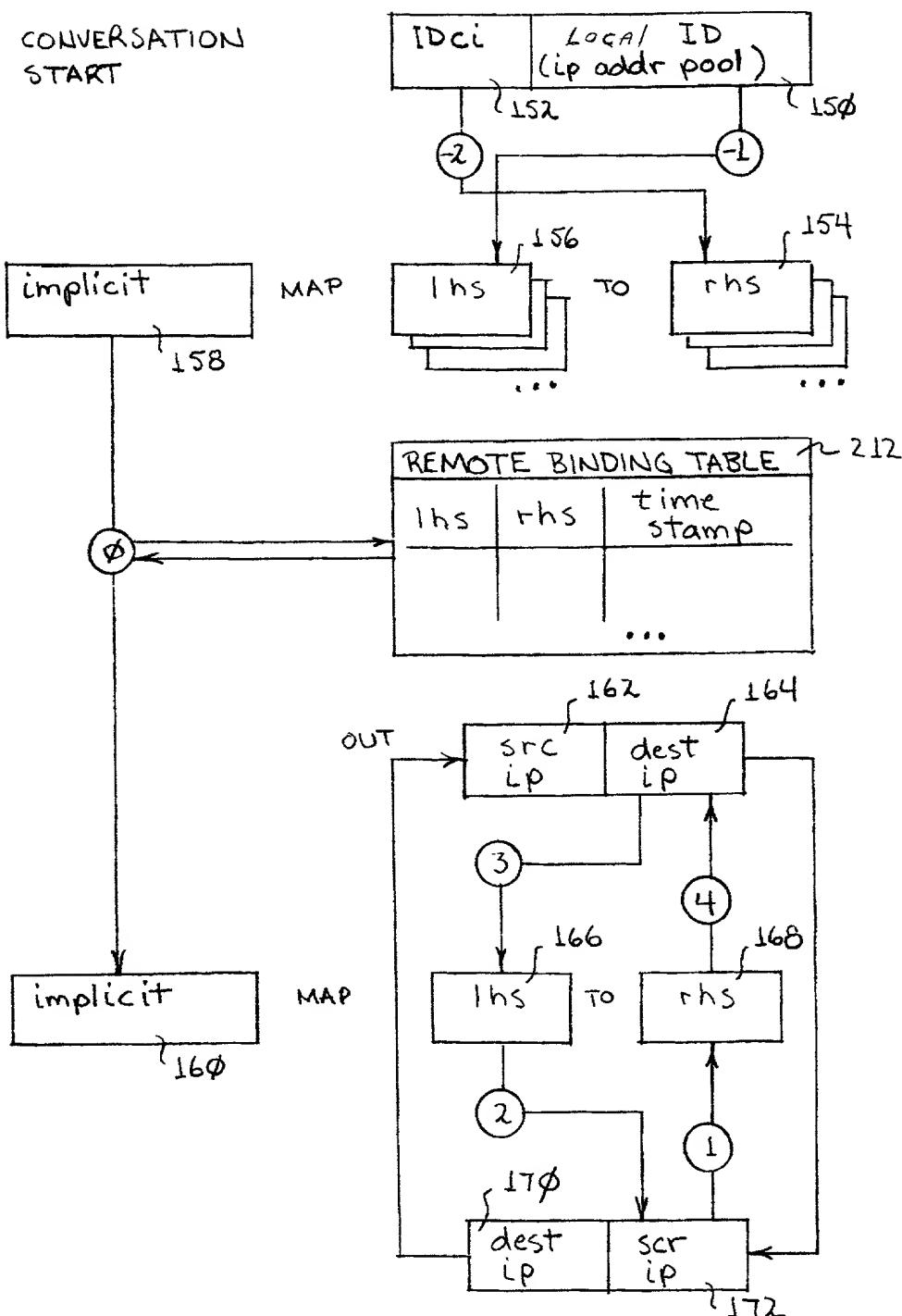


FIG. 6

END9 1999 φ129 US1
7/8

CONVERSATION
START

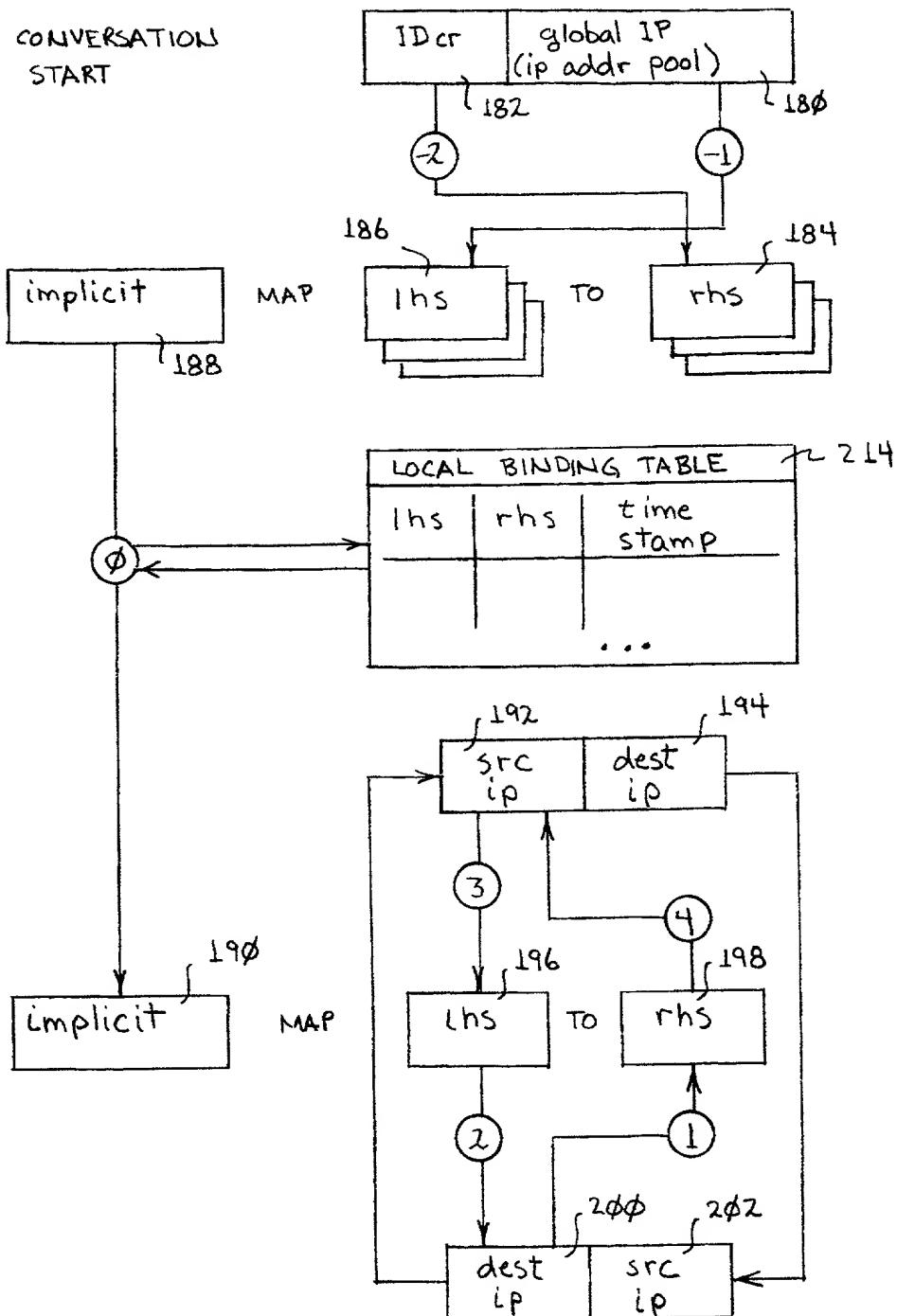


FIG. 7

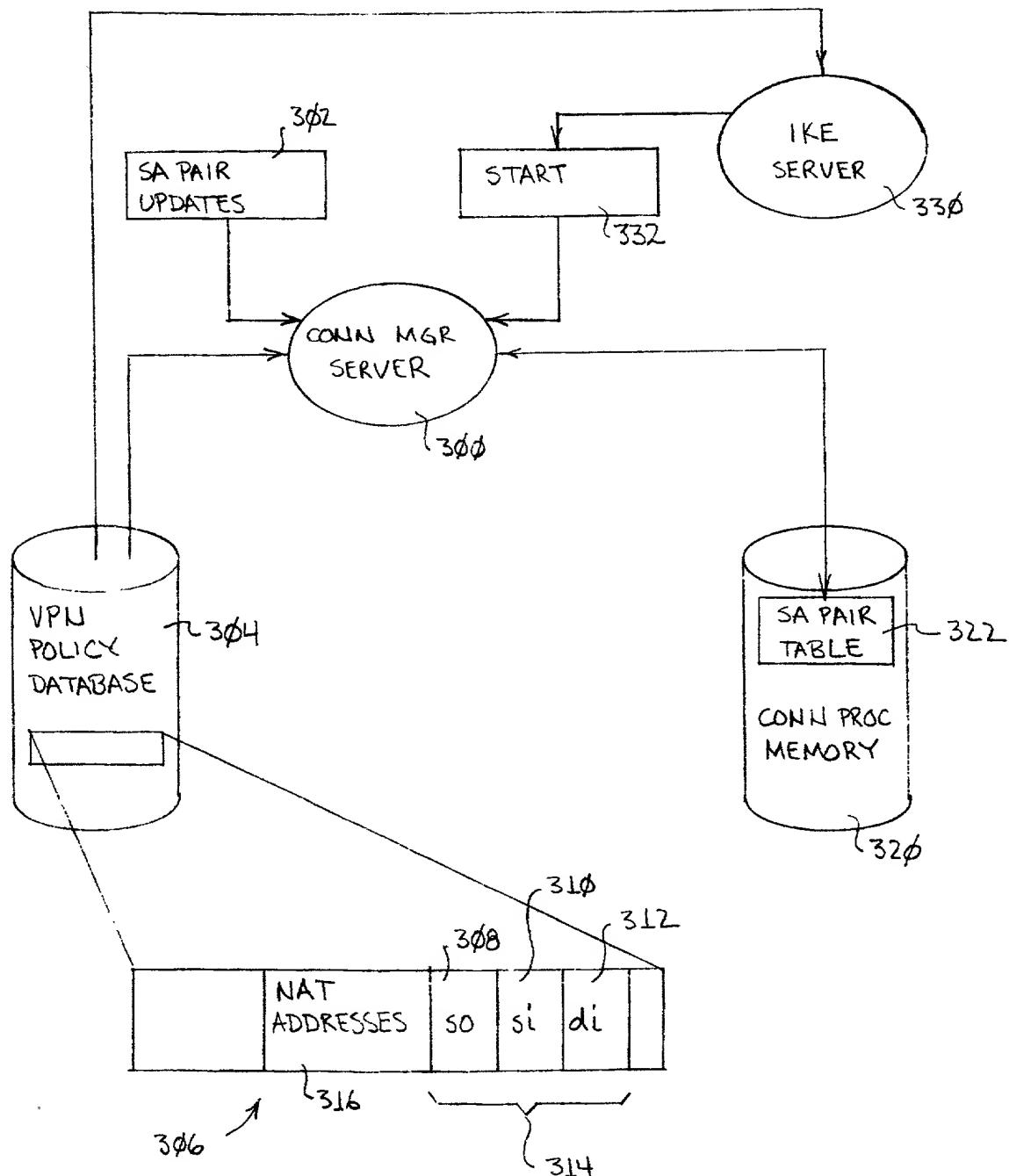


FIG. 8

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: **SYSTEM AND METHOD FOR NETWORK ADDRESS TRANSLATION INTEGRATION WITH IP SECURITY**

the specification of which (check one)

X is attached hereto.

was filed on _____ as Application Serial No. or PCAT International Application No. _____ and was amended on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below any foreign application for patent or inventors certificate, or PCAT International application having a filing date before that of the application on which priority is claimed::

Prior Foreign Appplication(s):

Number	Country	Date/Month/Year	Priority Claimed
--------	---------	-----------------	------------------

I hereby claim the benefit under 35 U.S.C. Section 119(e) of any United States provisional application(s) listed below.

Application Number	Filing Date
--------------------	-------------

I hereby claim the benefit under Title 35, United States Code, section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. Section 112, I acknowledge the duty to disclose information material to patentability of this application as defined in 37 CFR Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

Prior U.S. Applications:

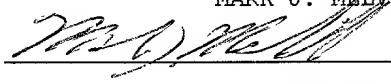
Serial No.	Filing Date	Status (patented, pending, abandoned)
09/240,720	1/29/99	Pending

As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith: David L. Adour, Reg. No. 29,604; Lawrence R. Fraley, Reg. No. 26,885; John R. Pivnichny, Reg. No. 43,001; Arthur J. Samodovitz, Reg. No. 31,297; William H. Steinberg, Reg. No. 28,540; Christopher A. Hughes, Reg. No. 26,914; Edward A. Pennington, Reg. No. 32,588; John H. Hoel, Reg. No. 26,279; Joseph C. Redmond, Jr., Reg. No. 18,753; and Shelley M Beckstrand, Reg. No. 24,886.

Send all correspondence to: Shelley M Beckstrand, P.C.
 Attorney at Law
 314 Main Street
 Owego, NY 13827

Direct telephone calls to: (607) 687-9913 [alt: (607) 755-3268]

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

- (1) Inventor: EDWARD B. BODEN
 Signature:  Date: 5/22/2000
- Residence: 3217 KNAPP ROAD, VESTAL, NY 13850
 Citizenship: USA
 Post Office Address: SAME AS RESIDENCE
- (2) Inventor: MARK J. MELVILLE
 Signature:  Date: 5/22/2000
- Residence: 1301 FOXBORO LANE, ENDWELL, NY 13760
 Citizenship: USA
 Post Office Address: SAME AS RESIDENCE
- (3) Inventor: TOD A. MONROE
 Signature:  Date: 5/22/2000
- Residence: 241 EDSON ROAD, ENDICOTT, NY 13760
 Citizenship: USA
 Post Office Address: SAME AS RESIDENCE
- (4) Inventor: FRANK V. PAXHIA
 Signature:  Date: 05/22/2000
- Residence: 1111 AIRPORT ROAD, BINGHAMTON, NY 13905
 Citizenship: USA
 Post Office Address: SAME AS RESIDENCE